

## Attorney Docket No. SYN006-05

**Amendments to the Specification**

Please amend the specification as follows:

On page 27, please replace the paragraph beginning on line 13 with following amended paragraph:

The handling of a failure of one of the services in a peer group is relatively trivial. The failure might be detected when a failed Foo service does not renew its lease with the group service, or when the client's group proxy detects that a failed Foo did not provide a response to an invocation and then notifies the group service 24. The failed Foo's service proxy is simply removed from the peer group [[proxy]] logic shells at the clients 2 and the look-up service 20 bundle as described above with respect to Figure 8. In a peer group configuration, the transition period is much short than for a CC group, so buffering may not be needed.

On page 32, please replace the paragraph beginning on line 7 with following amended paragraph:

In the described embodiment, because there are different group logic shells for peer and CC modes, the group proxy is completely replaced when transitioning between modes. In an alternative embodiment a single group logic shell can contain the logic for both a CC group and a peer group. With a single group logic shell, in order to switch modes, the group service only has to update the service proxies bundled within group logic shell at the clients and look-up

**Attorney Docket No. SYN006-05**

service. Specifically, in a switch from peer mode to CC mode, the group service would select one service and instruct it (or its grouping agent) to become the coordinator. The group service would then announce to all fielded group proxies that are distributed at clients that they should delete all of the service proxies except the coordinator's service proxy from the group logic shell. In a switch from CC mode to peer mode, the group service would add the necessary service proxies to the multiple instances of the group [[proxy]] logic shells at the clients to form a new group proxy. In both cases, the group service would still unregister and reregister the group with the look-up service to reflect the group's new incarnation. One advantage to this embodiment is that there is no need to handshake and pass buffered commands from one group proxy to another, since the same group proxy remains in place at each client.

Please replace the abstract with the following paragraph:

The invention is a method of handling ~~a wide range of dynamic~~ groups of services where the makeup of the groups can be determined and changed while the application is running. This is mainly accomplished through a group proxy, which is generated at run time, and which handles interactions with groups of services on behalf of one or more clients. The group proxy consists of a group logic shell which contains all the group-aware logic, and a service proxy for each service in the group which contains the necessary logic to interact with the particular service. The group proxy, which is given to a client for all of its interactions with the group of services, buffers calls from that client to its group

**Attorney Docket No. SYN006-05**

when the group is unavailable because it is in transition. When the transition is complete the group proxy transmits the stored client commands to the group. ~~In the preferred embodiment of the invention, all the group-aware logic for a distributed computing application is provided in separate code modules, namely the group proxy, group service and grouping agent, thus relieving clients and services from providing this logic and maintaining the purity of the look-up service and other infrastructure services.~~